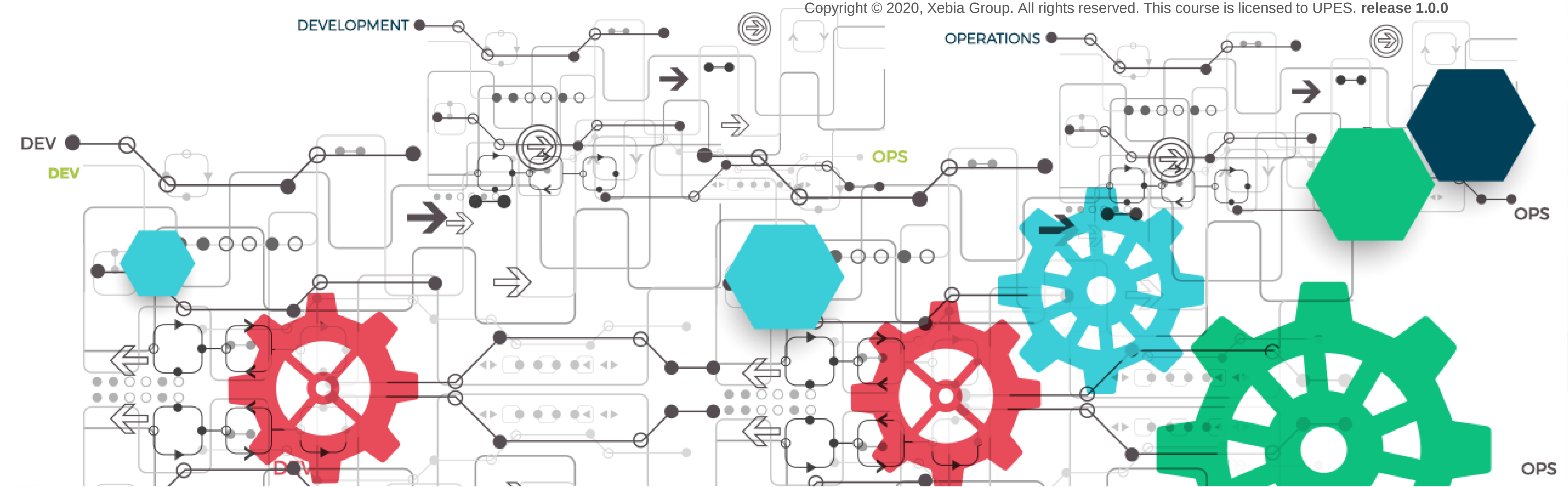
Semester **06**

B.TECH CSE with Specialization in DevOps

Application Containerization

Module # 01

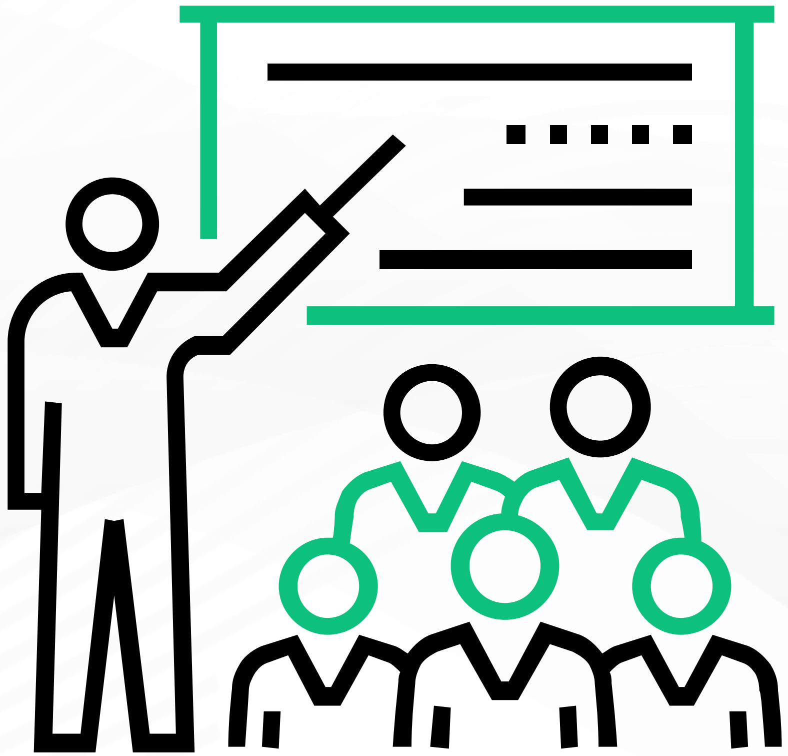
Understanding Containers



Module Objectives

At the end of this module, you will be able to:

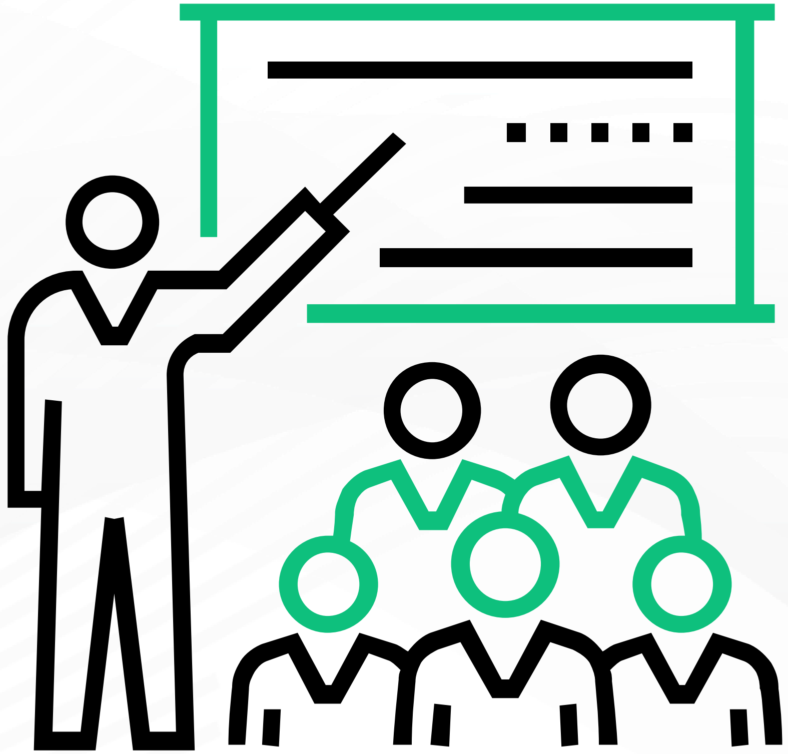
- Define transporting "goods" analogy
- Explain virtualisation and comparison with virtual machines
- Describe containerization platform, images and runtime
- Identify the journey of container technology
- Discuss the chroot system call
- Explain FreeBSD Jails, LinuX Containers (LXC) and Docker



Module Topics

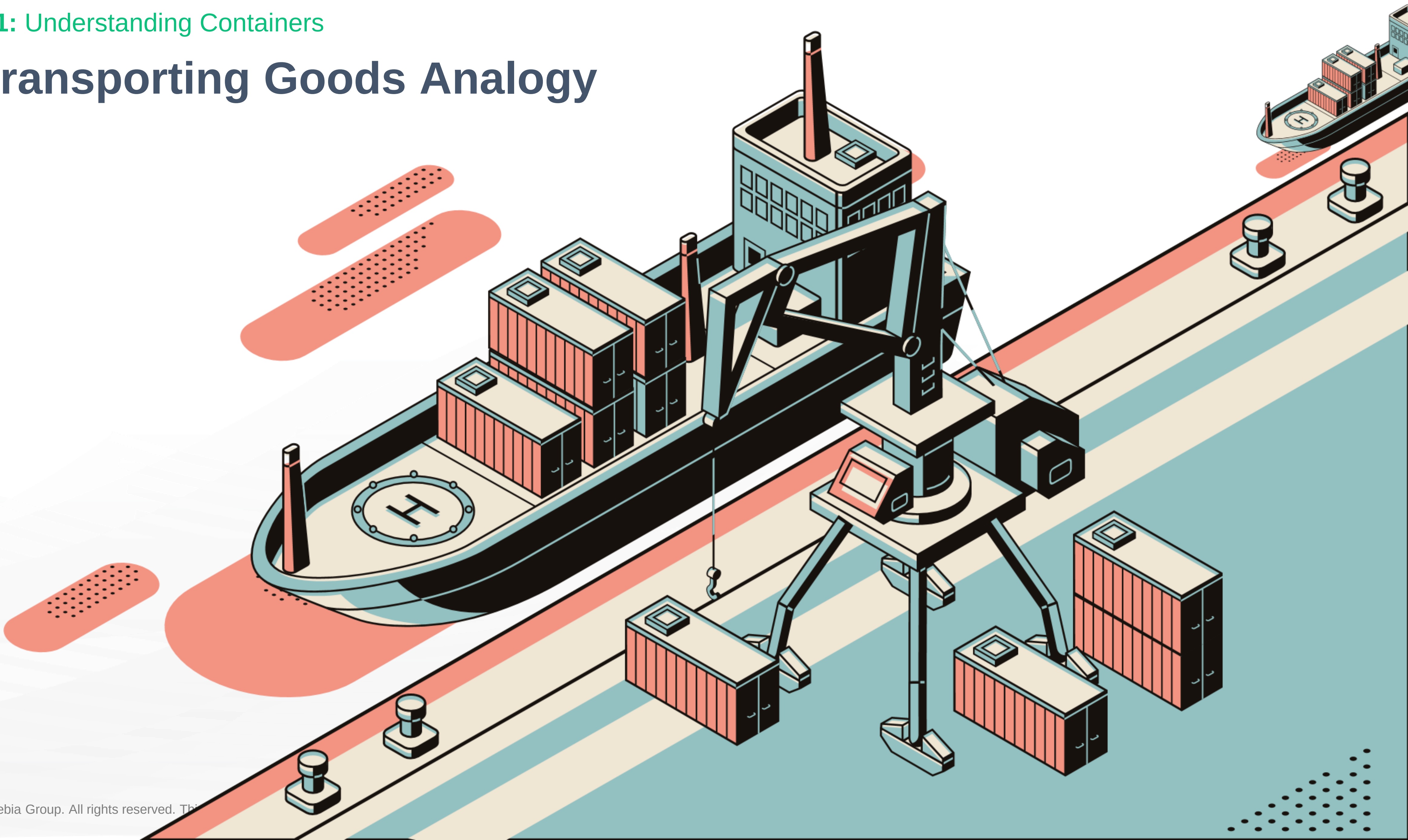
Let us take a quick look at the topics that we will cover in this module:

- ⇥ Transporting "goods" analogy
- ⇥ Containerization platform, images and runtime
- ⇥ Comparison with virtual machines
- ⇥ The chroot system call
- ⇥ FreeBSD Jails
- ⇥ LinuX Containers (LXC)
- ⇥ Docker



Module 1: Understanding Containers

1.1 Transporting Goods Analogy



1.2 Problems in Shipping Industry before Containers

The shipping industry faced the following problems before containers:

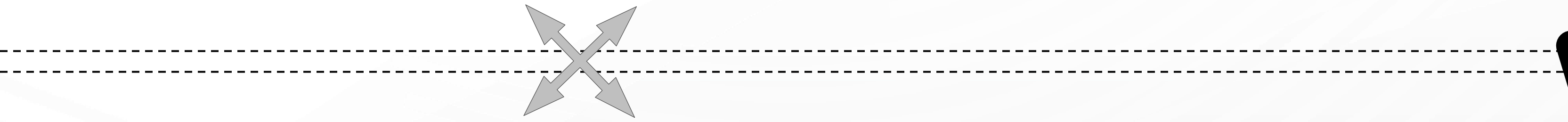
- Shipping fragile goods with robust ones
- Shipping of edible food items with raw materials
- Shipping of cars
- Shipping of various goods via different mode of transports like railways, roadways, airways, waterways, etc.
- Loading and unloading of goods

1.3 Shipping Industry Challenges

The various challenges faced by the shipping industry.

Multiplicity of goods

Do I worry about how goods interact? (e.g., coffee beans next to spices)



Multiplicity of methods for transporting/storing

Can I transport quickly & smoothly? (e.g., from boat to train to truck)

Module 1: Understanding Containers

1.4 Container: The Saviour

How did the container become the saviour?

Multiplicity of goods

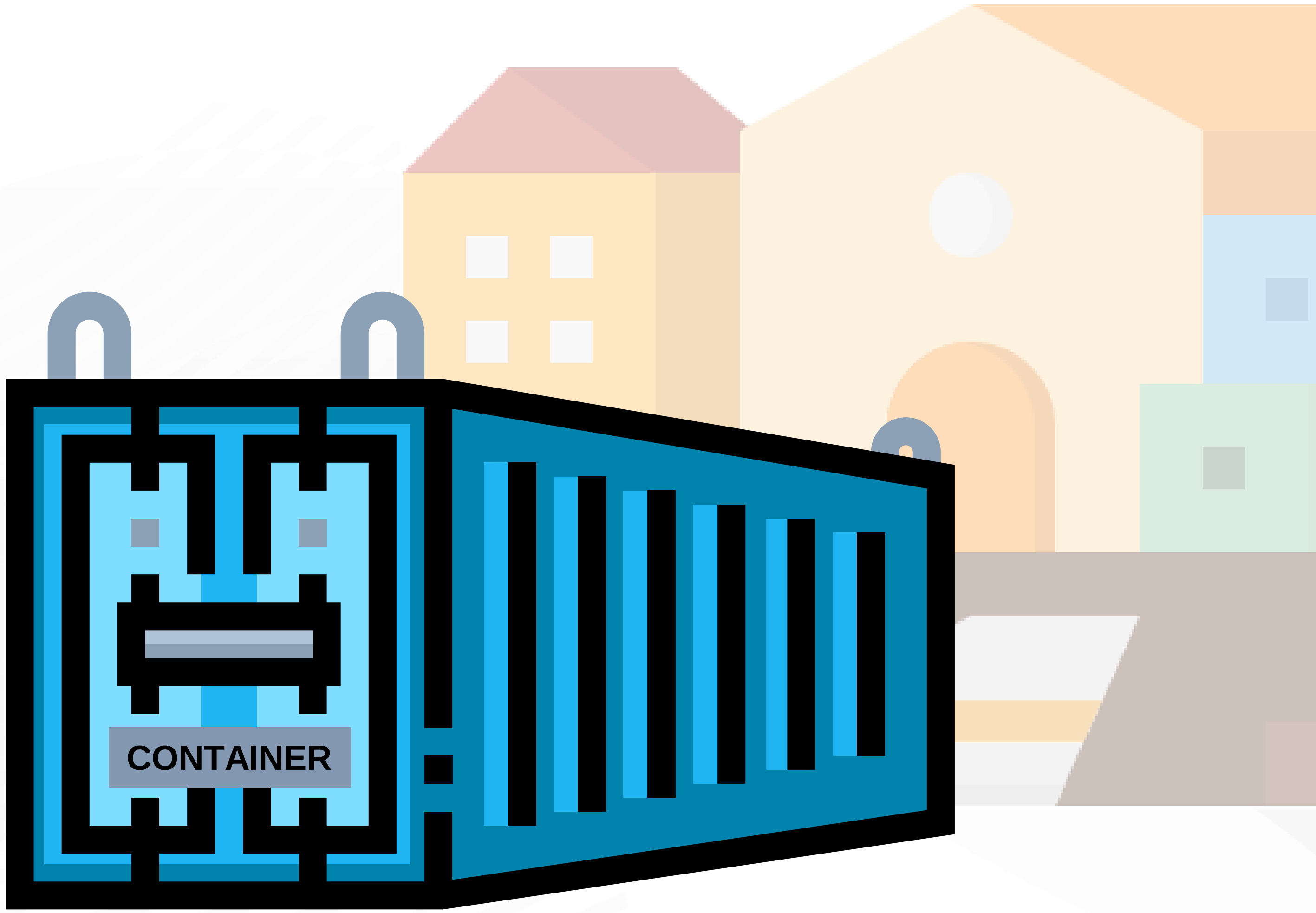
Do I worry about how goods interact? (e.g., coffee beans next to spices)

A standard container that is loaded with virtually any goods, & stays sealed until it reaches final delivery.

Multiplicity of methods for transporting/storing

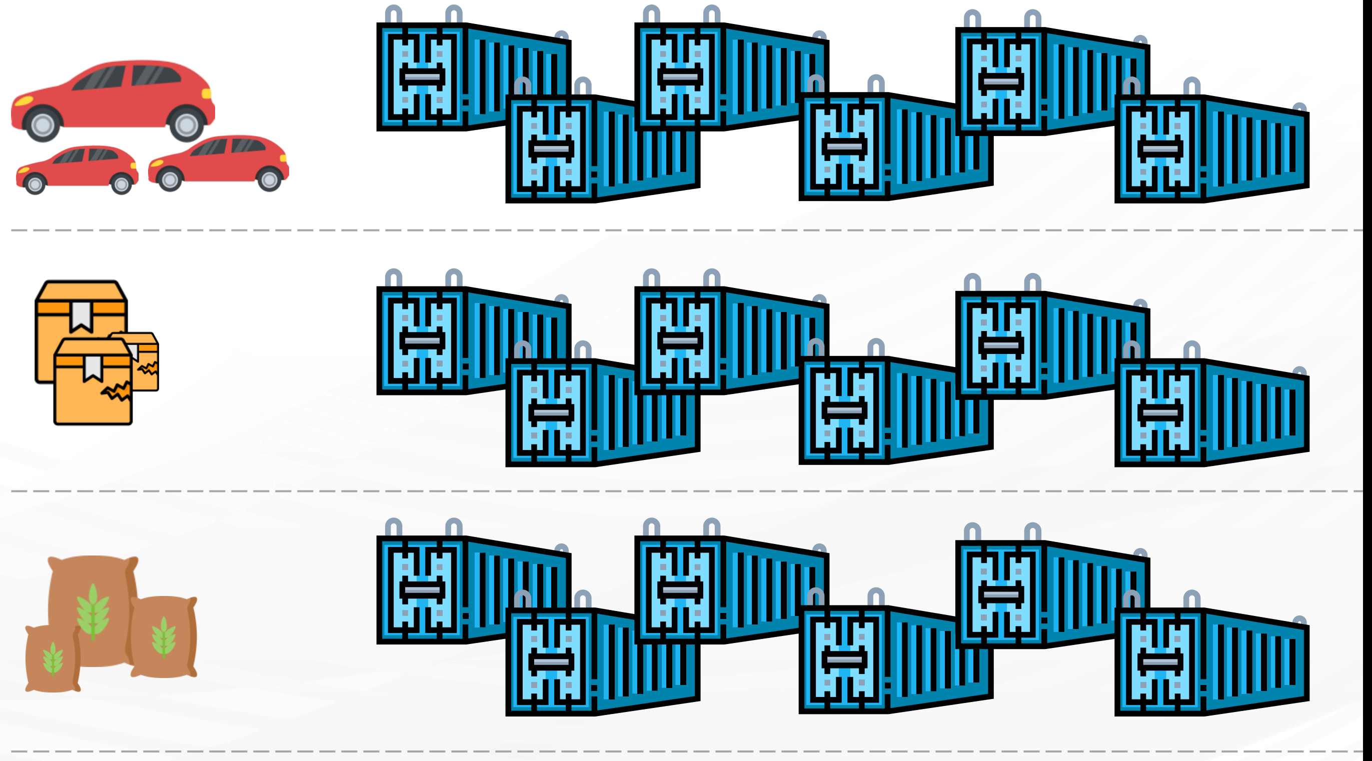
Can I transport quickly & smoothly? (e.g., from boat to train to truck)

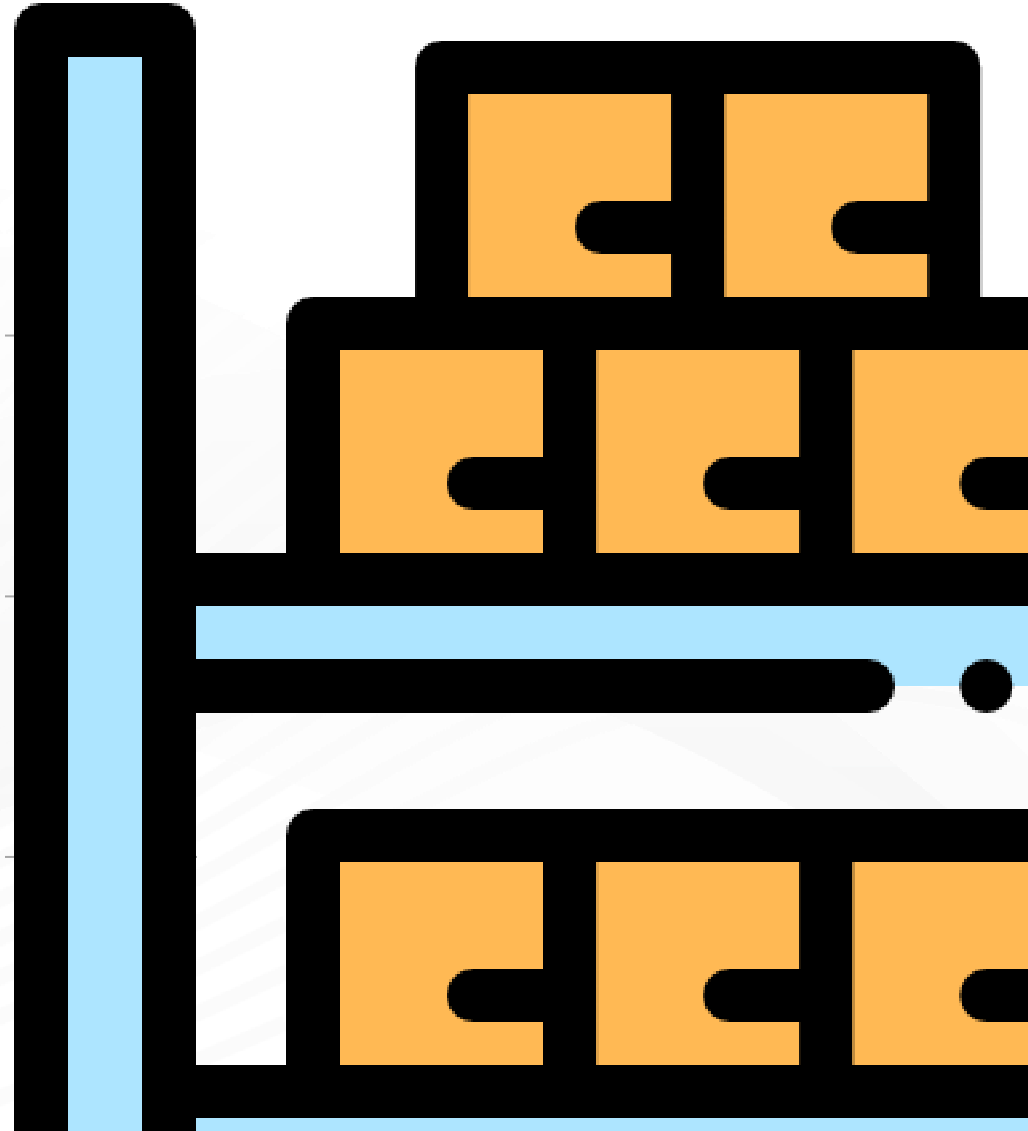
In between, can be loaded & unloaded, stacked, transported efficiently over long distances, & transferred from one mode of transport to the other.



1.5 Solution by Containers in the Shipping Industry

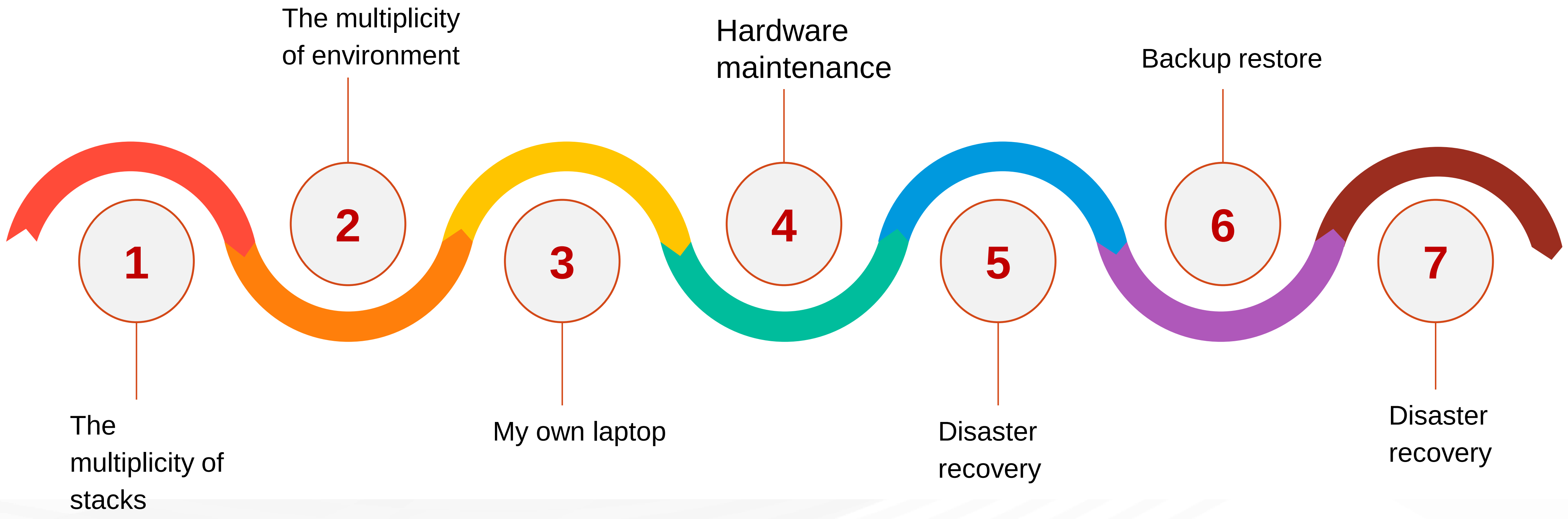
Everything falls into place with the help of containers.





1.6 Challenges in the Software Industry

The various challenges in the software industry are as follows:



1.6 Challenges in the Software Industry (Contd.)

The various challenges in the software industry are as follows:

Do services & apps interact appropriately? **Multiplicity of Stacks**

Static Website
nginx 1.5, modsecurity, openssl, bootstrap2

User DB
postgresql, pgv8, v8

Web frontend
Ruby, Rails, sass, Unicorn

Background workers
Python 3.0, celery, pyredis, libcurl, ffmpeg, libopencv,nodejs, phantomjs

API Endpoint
Python 2.7, Flask, pyredis, celery, psycopg, postgresql-client

Queue
Redis, redis-sentinel

Can I migrate smoothly & quickly? **Multiplicity of hardware environments**

Development VM **Customer Data Center** **Contributor's laptop** **Public Cloud**

QA server **Production cluster** **Data recovery** **Production Servers**

1.7 Problems in Software Industry Before Containers

The chaos in the software industry while managing diverse stack in different environments:

	Development VM	QA Server	Single Prod Sever	Onsite Cluster	Public Cloud	Contributor Laptop	Customer Servers
Static Website	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
Web Front End	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Module 1: Understanding Containers

1.7 Problems in Software Industry Before Containers (Contd.)

The chaos in the software industry while managing diverse stack in different environments:

	Development VM	QA Server	Single Prod Sever	Onsite Cluster	Public Cloud	Contributor Laptop	Customer Servers
Static Website	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
Web Front End	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Module 1: Understanding Containers

1.8 Put that in Container!

Multiplicity of stacks

Developer: Build once, run anywhere(finally)

Static Website
nginx 1.5, modsecurity, openssl, bootstrap2

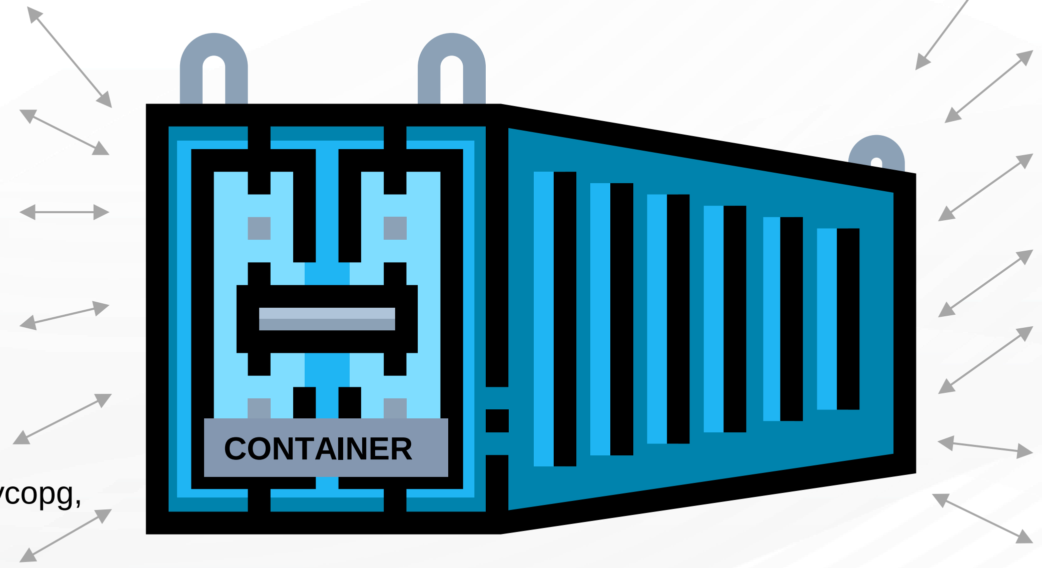
User DB
postgresql, pgv8, v8

Web frontend
Ruby, Rails, sass, Unicorn

Background workers
Python 3.0, celery, pyredis, libcurl, ffmpeg, libopencv,nodejs, phantomjs

API Endpoint
Python 2.7, Flask, pyredis, celery, psycopg, postgresql-client

Queue
Redis, redis-sentinel



Multiplicity of hardware environments

Operator: Configure once, run anything

Development VM

Production Servers

Customer Data Center

Contributor’s laptop

QA server

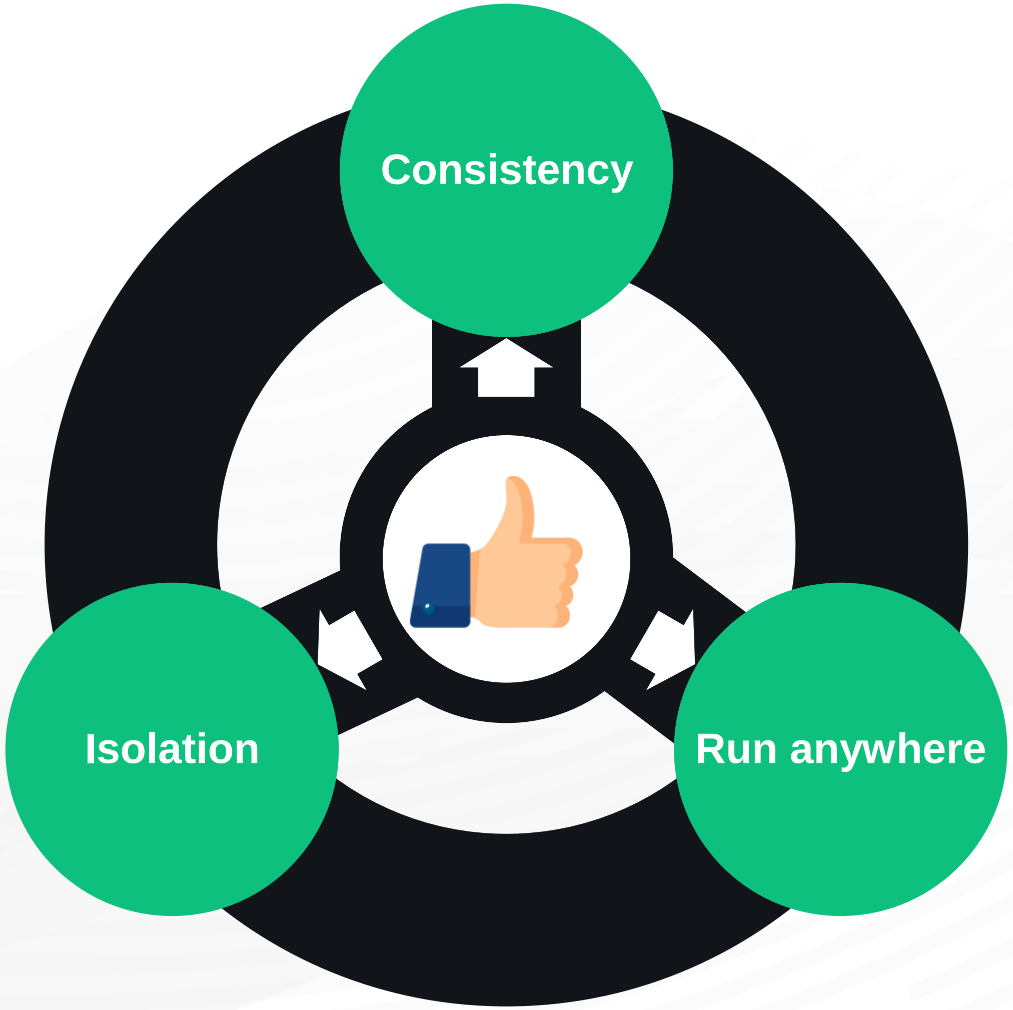
Production cluster

Data recovery

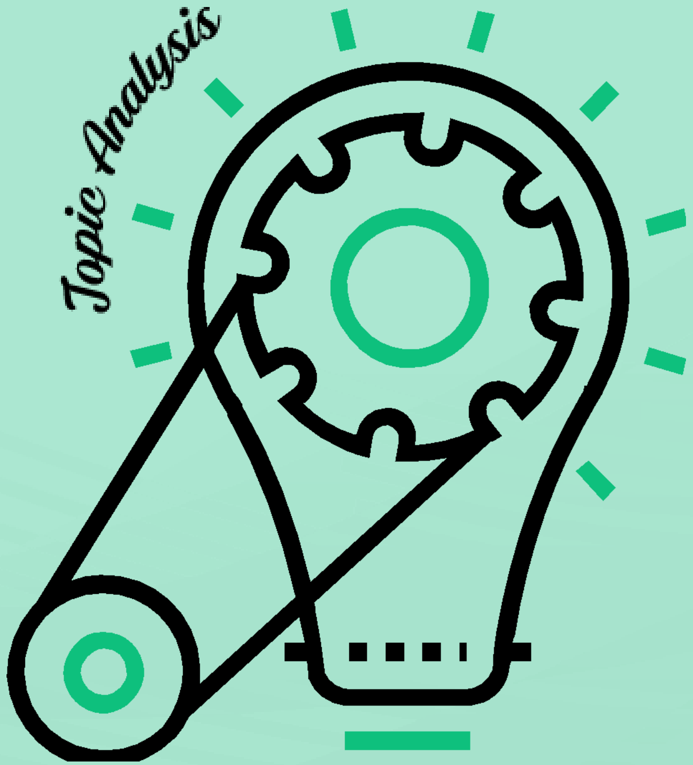
Source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRWHcB-4JmViuSdW3tEtckQ2HLOrlhv-AnO9_6jzUohl1NmPyOV&s

1.9 Solution by containers in the Software Industry

Keeping everything in container solves our problems easily on the factors below:



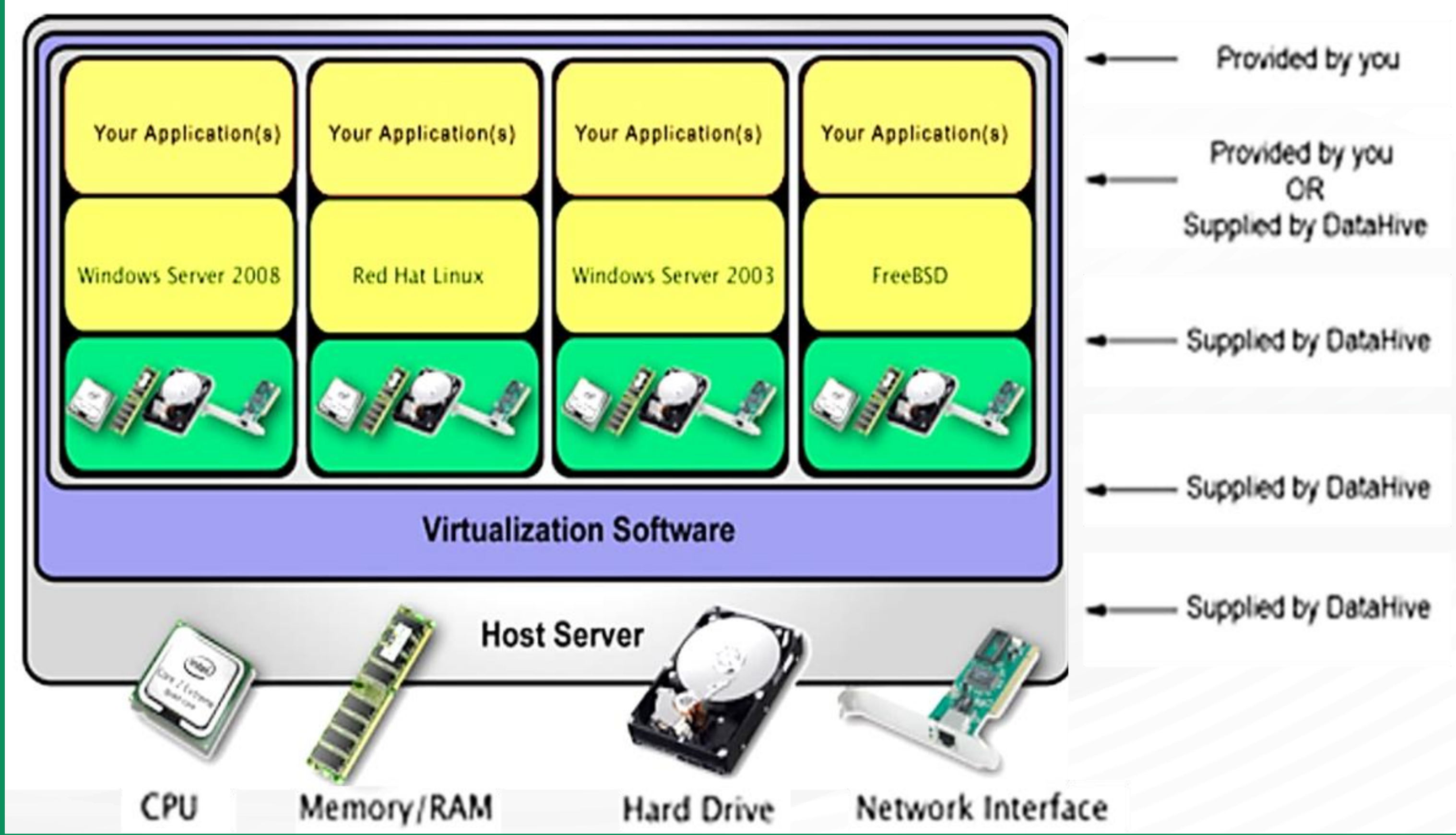
What did you Grasp?



1. Which is the key-factor which helped shipping and software industry to overcome the problems?
   - **A) Isolation**
   - **B) Availability**
   - **C) Consistency**
   - **D) Performance**

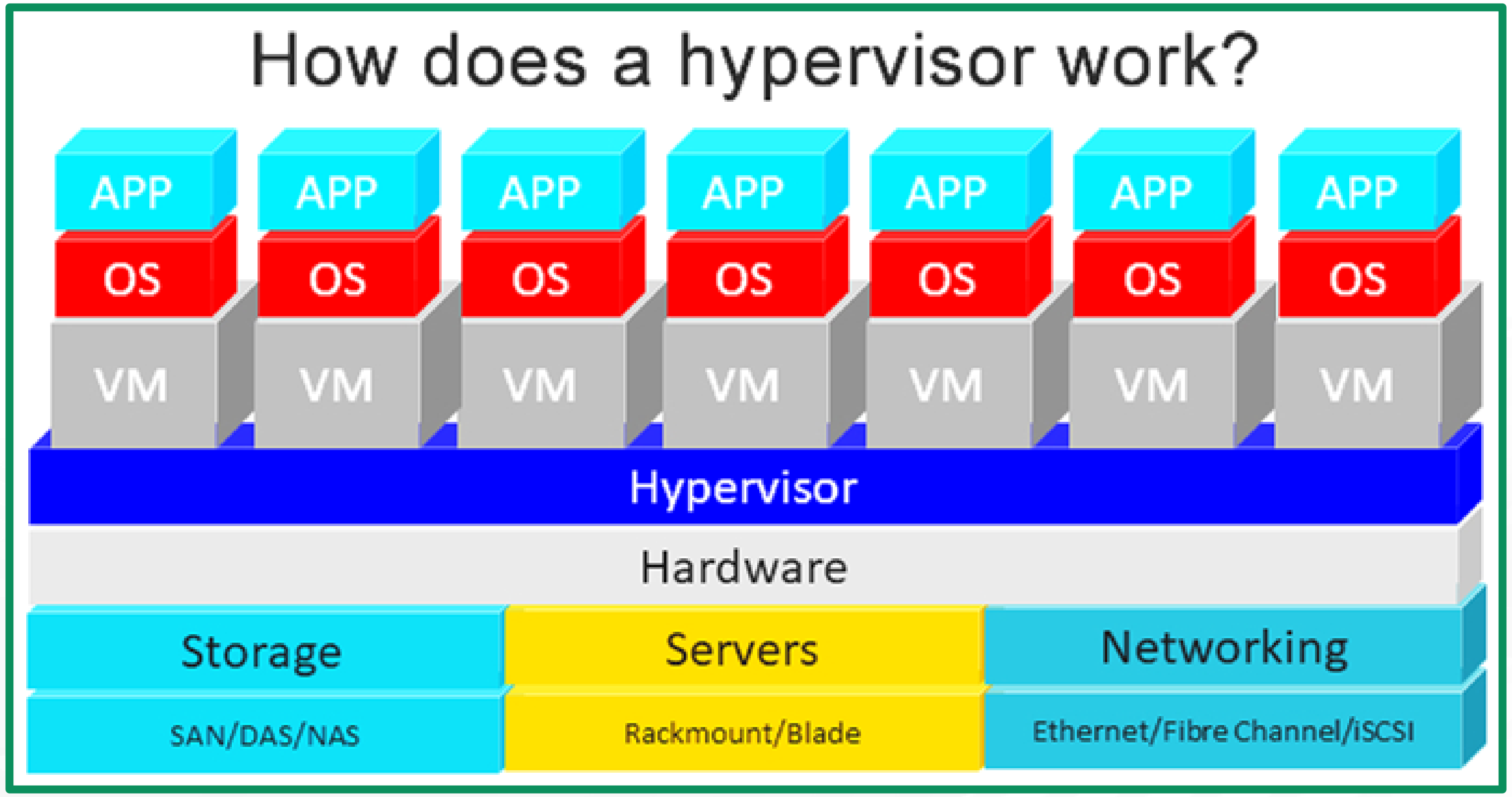
1.10 Virtualisation

Virtualization is a technology to run multiple same or different operating systems which are completely isolated from each other.



- ☐ Consolidation
- ☐ Shared CPU, Mem, Disk, NIC
- ☐ Maximum resource utilization
- ☐ Centralized Management
- ☐ Migration
- ☐ Less TCO (total cost of ownership)

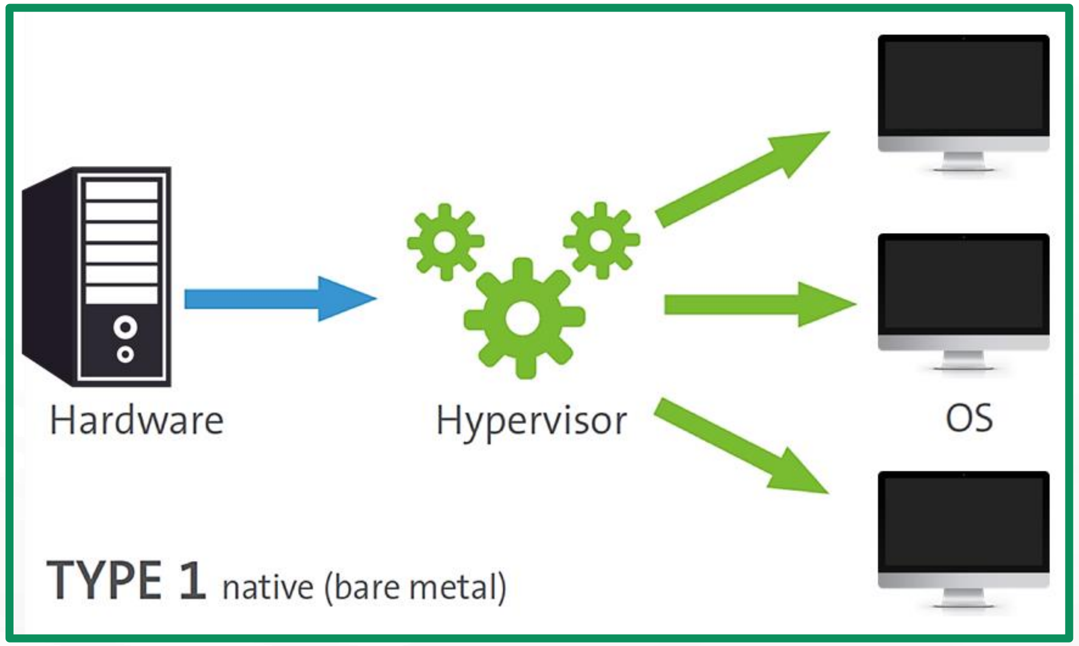
1.11 What is Hypervisor?

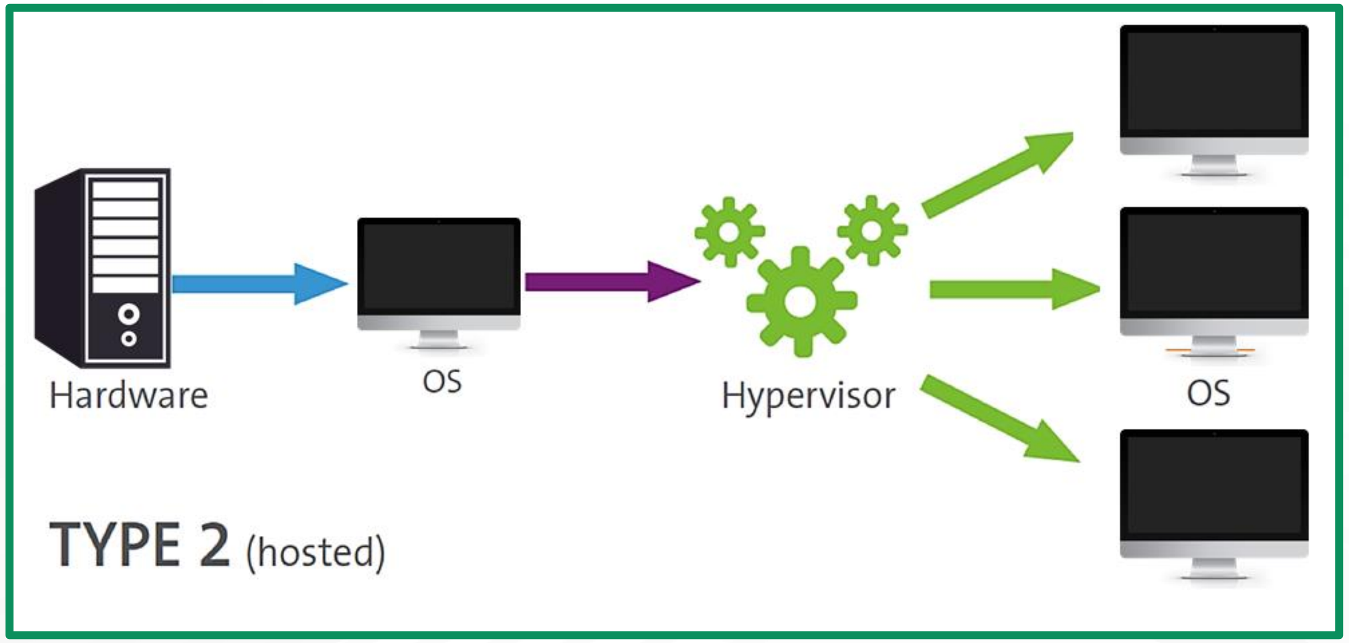


Source: https://www.data-storage.uk/articles/how-does-a-hypervisor-work/

1.12 Types of Hypervisors

The two types of hypervisors are as follows:

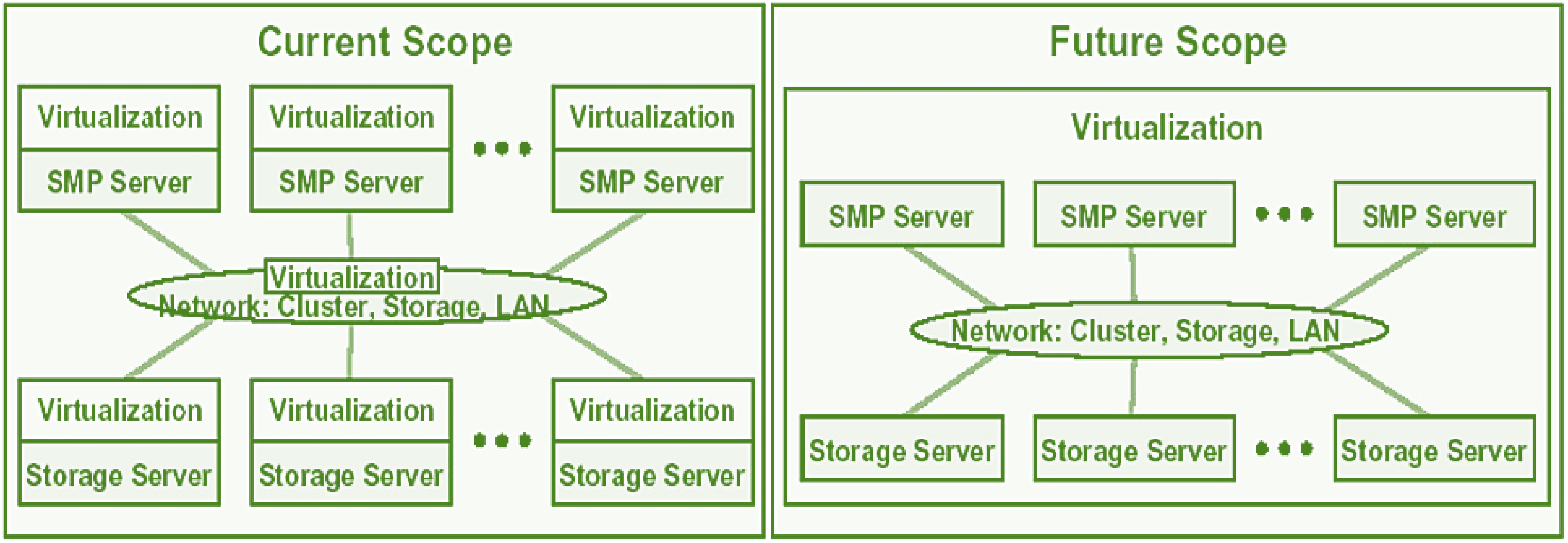




Source: https://www.flexiant.com/2014/02/05/what-does-a-hypervisor-do/

1.13 Scope of Virtualisation

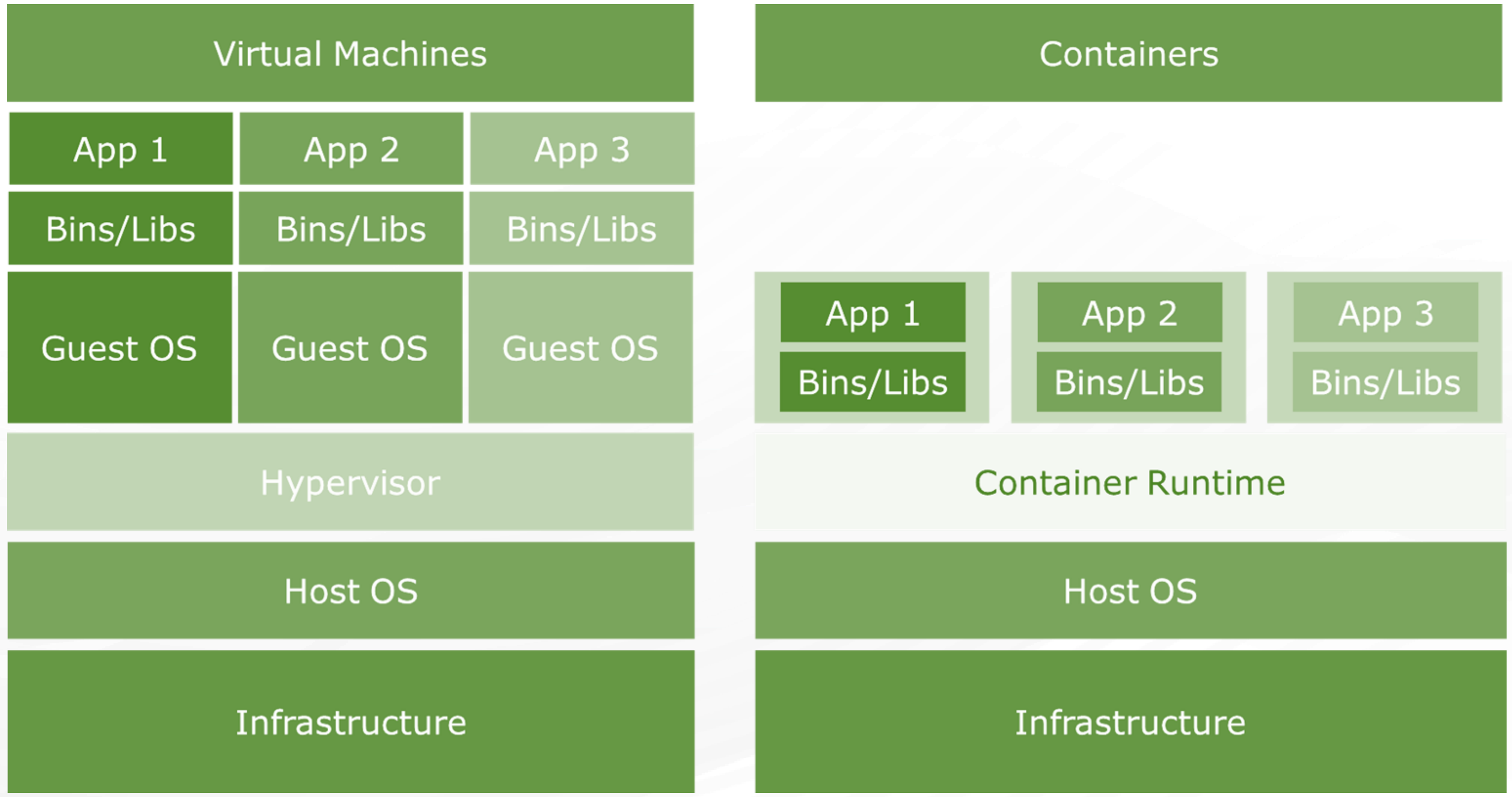
Virtualization has a wide scope in the Software Industry in terms of all the components required to keep our whole ecosystem up and running.



Source: https://www.slideshare.net/VaibhavSawant1/green-computing-1-2

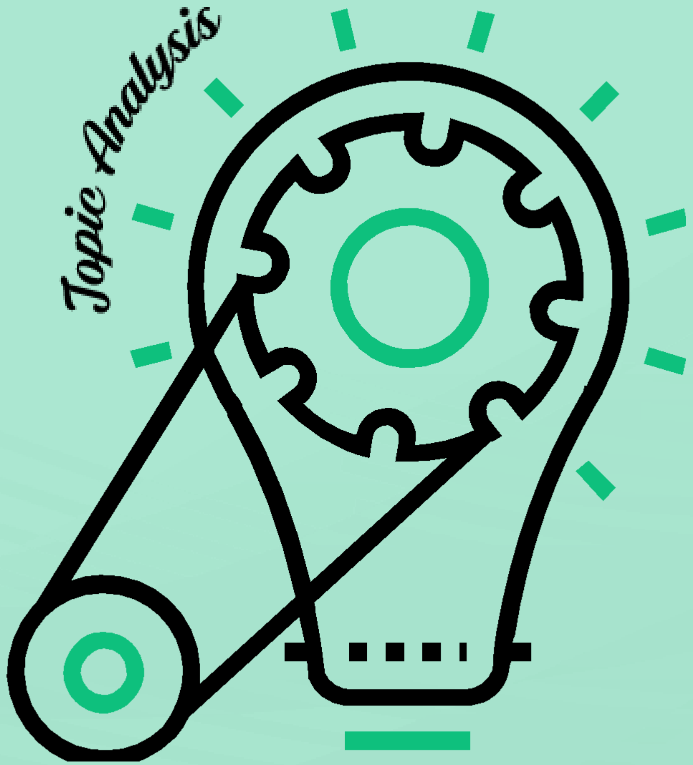
1.14 Containers vs Virtual Machines

Let’s look at the differentiation between containers and machines.



Source: https://i2.wp.com/www.dvoconsult.com/wp-content/uploads/2019/04/containers-vs-virtual-machines.jpg?fit=1024%2C536&ssl=1

What did you Grasp?

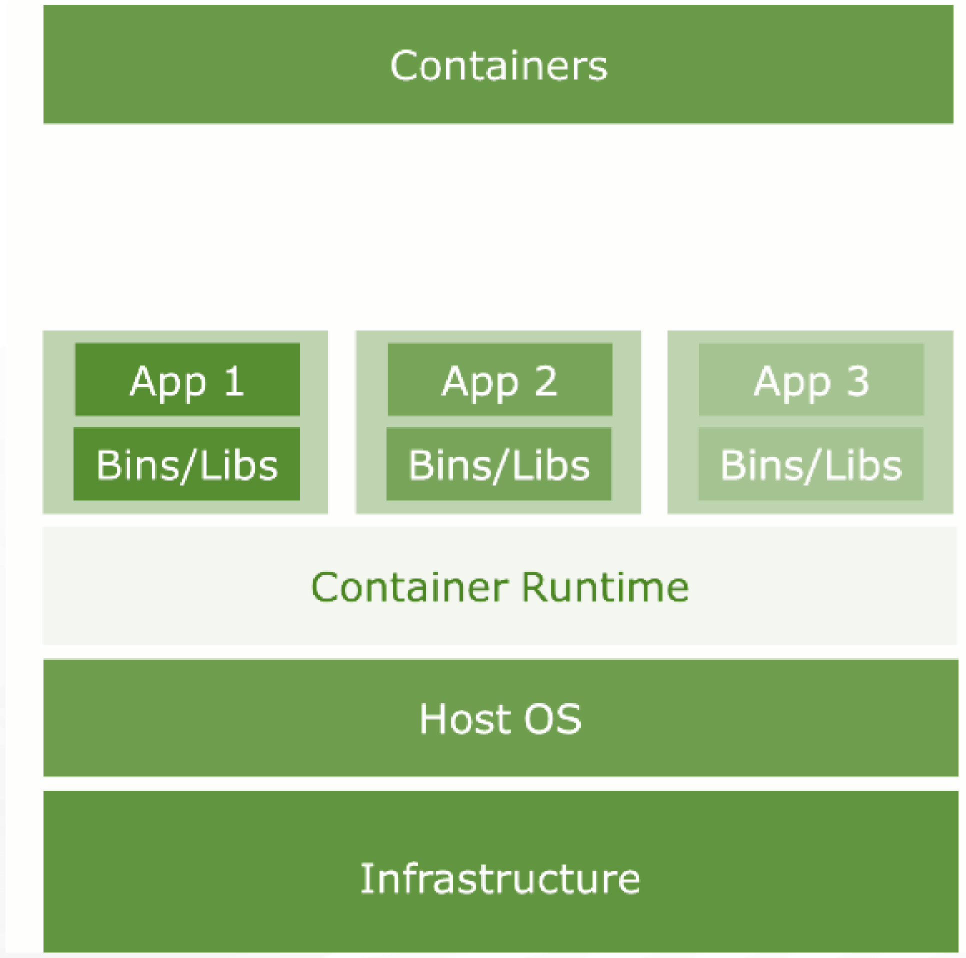


1. What is the main difference between Containerisation and Virtualization?
   - **A) Lifecycle**
   - **B) Isolation**
   - **C) Extraction at software level**
   - **D) Extraction at hardware level**

Module 1: Understanding Containers

1.15 Containerisation Platform, Runtime and Images

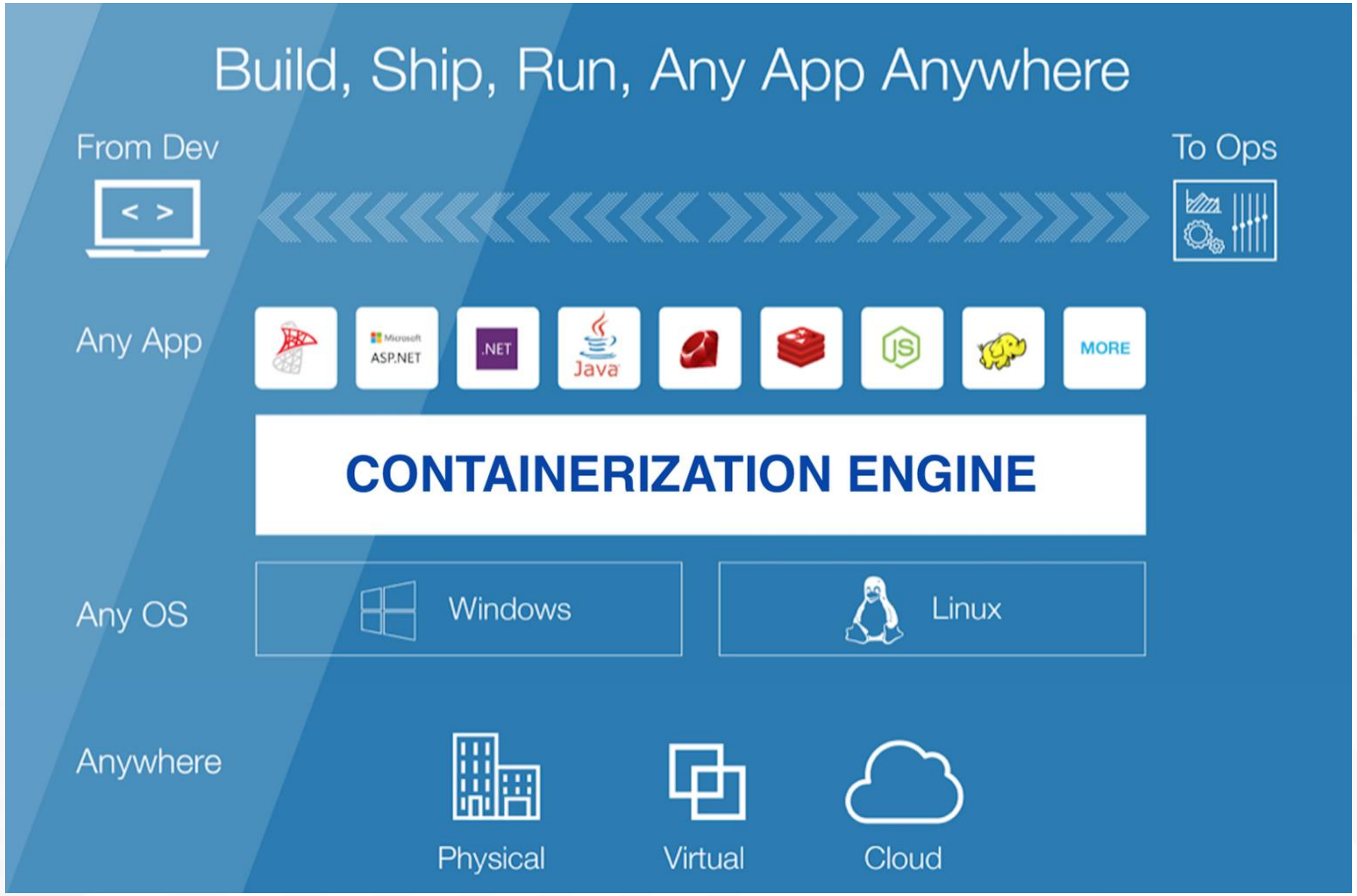
Let’s look at the containerisation ecosystem.



Source: https://i2.wp.com/www.dvoconsult.com/wp-content/uploads/2019/04/containers-vs-virtual-machines.jpg?fit=1024%2C536&ssl=1

1.16 Container Platform

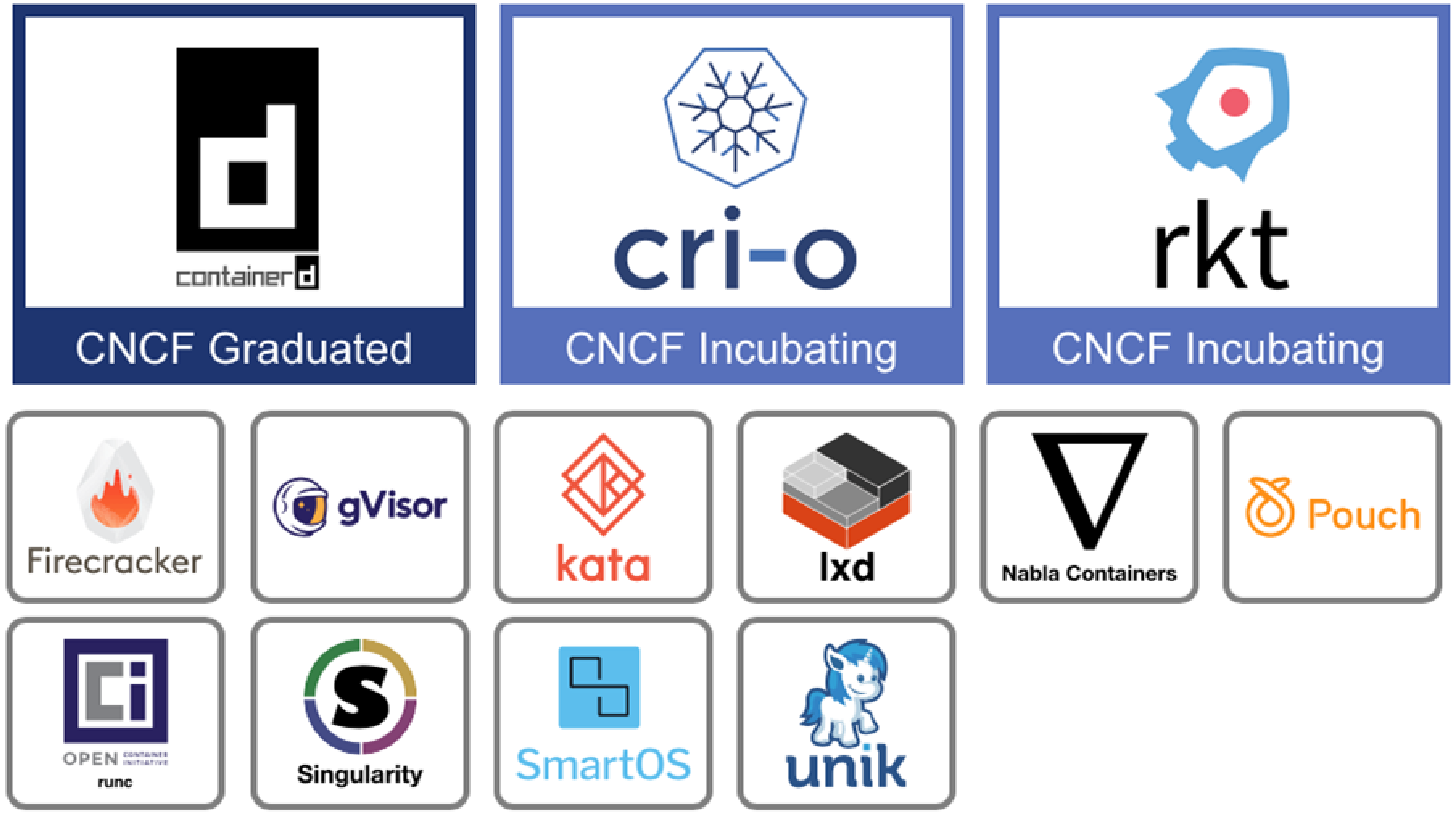
Containerization platform: It can be defined as a technology to isolate processes from each other, in such a manner that processes run like they are running in a normal operating system which is enforced by the container runtime.



Source: https://1.bp.blogspot.com/-kx-LtOGIwqU/XOzPWiztcVI/AAAAAAAAIvE/qURId5PRX40aV-TFJ-B7GA8t5CqAUF_3gCEwYBhgL/s1600/docker.png

1.17 Container Runtime

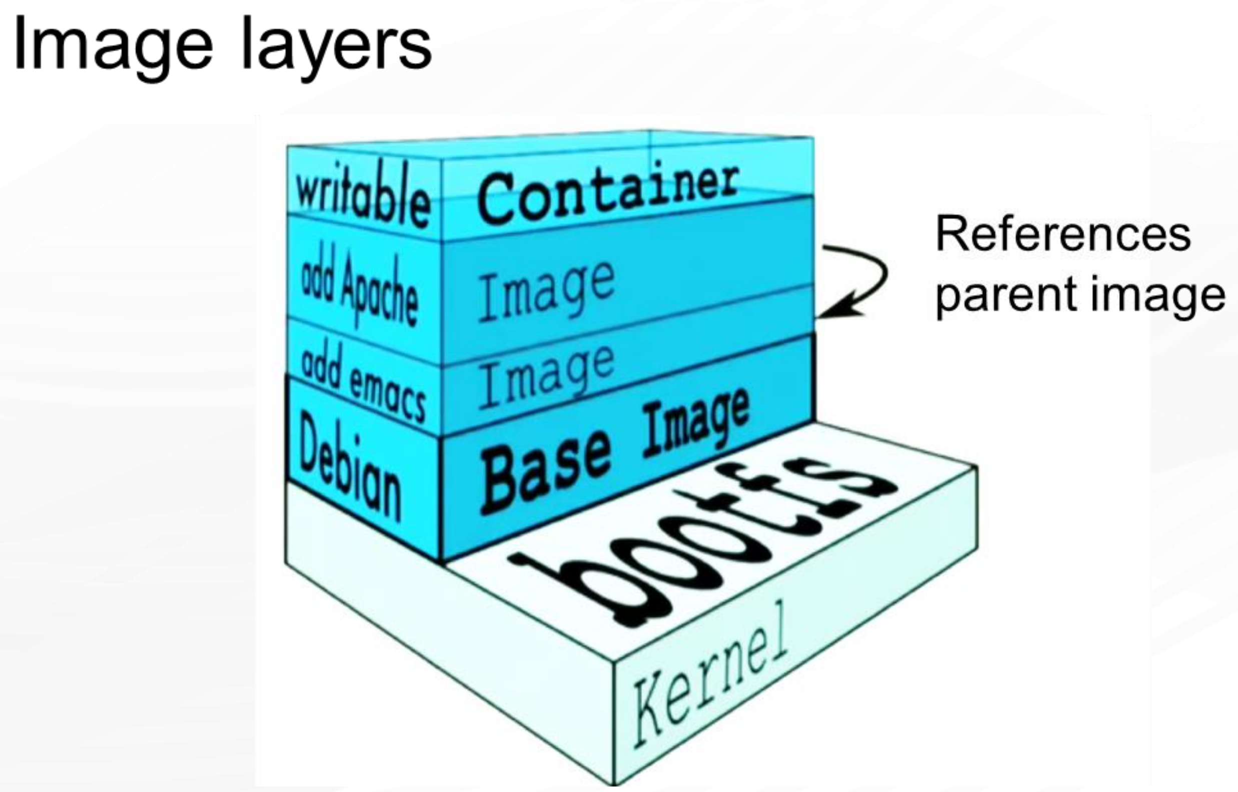
Container runtime: It is a Container execution environment, which ensures the allocation of limited shares of resources (e.g., CPU, memory, disk) to the containerized application, also exposes the services and APIs and tools for managing containers.



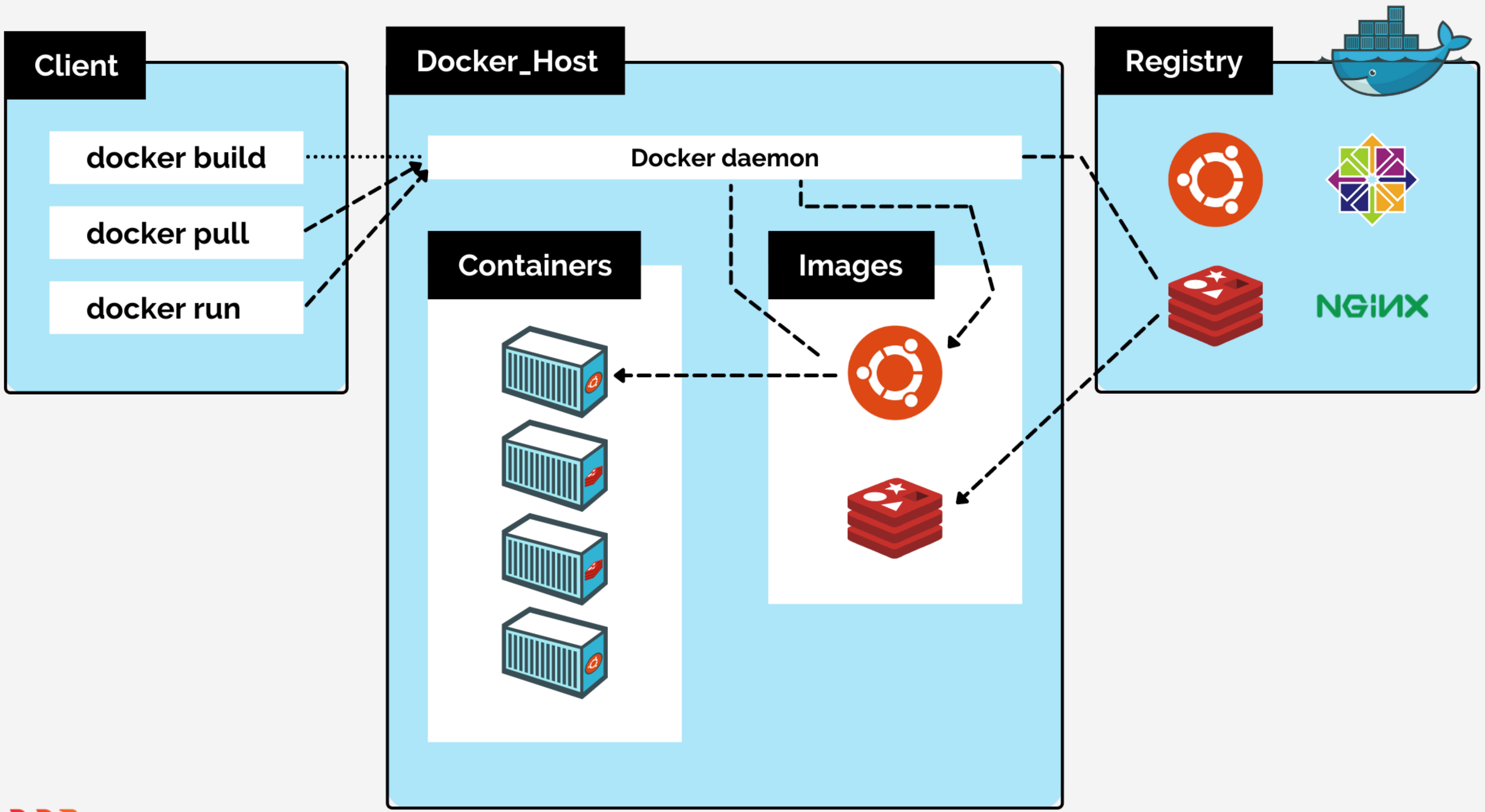
*Source: https://miro.medium.com/max/2486/1*OnB-Ah-FehjzQfhu5-BHmQ.png*

1.18 Container Images

Image: Images are readable files which will be used to spin up containers. An image defines the file system and execution parameters for the container. Images can be layered, composable, depending on the format of the runtime.



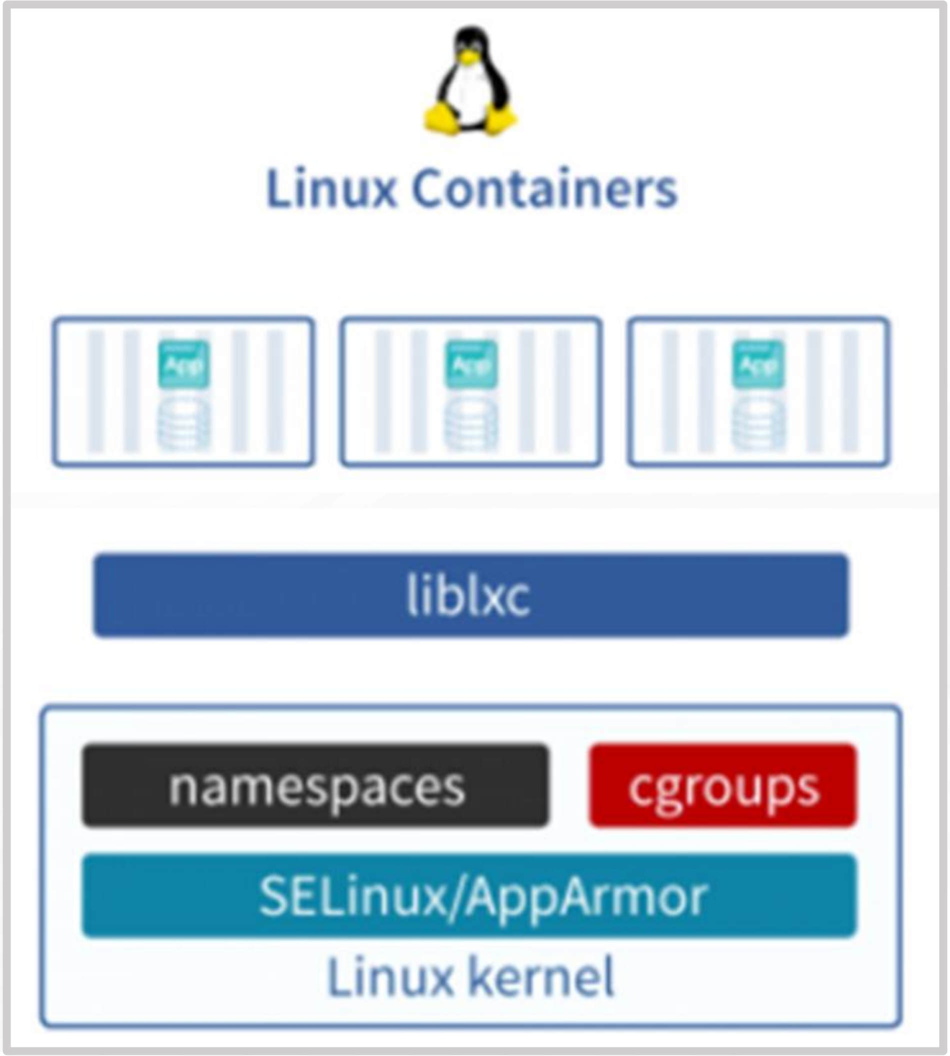
Source: https://s3.amazonaws.com/dev.assets.neo4j.com/wp-content/uploads/20160112150631/image-layers-docker.png


Client
docker build
docker pull
docker run
Docker_Host
Docker daemon
Containers
Images
Registry
NGINX

WHIZLABS

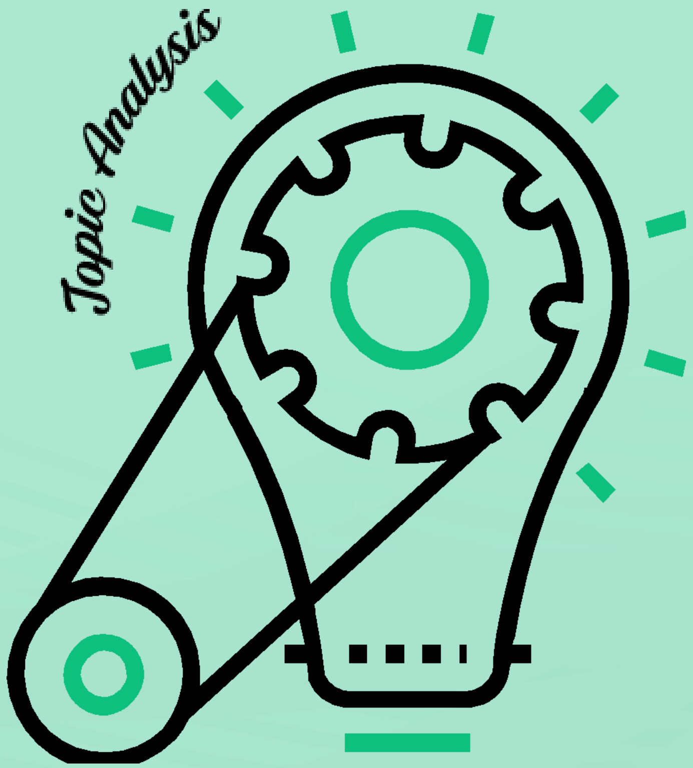
1.22 LinuX Containers (LXC)

Let’s look at the LinuX containers (LXC):



Source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR2oxv0OImn_QPxFahTuelvfsEz-mYpqUmo8zsVMyHcbTYZn04raA&s

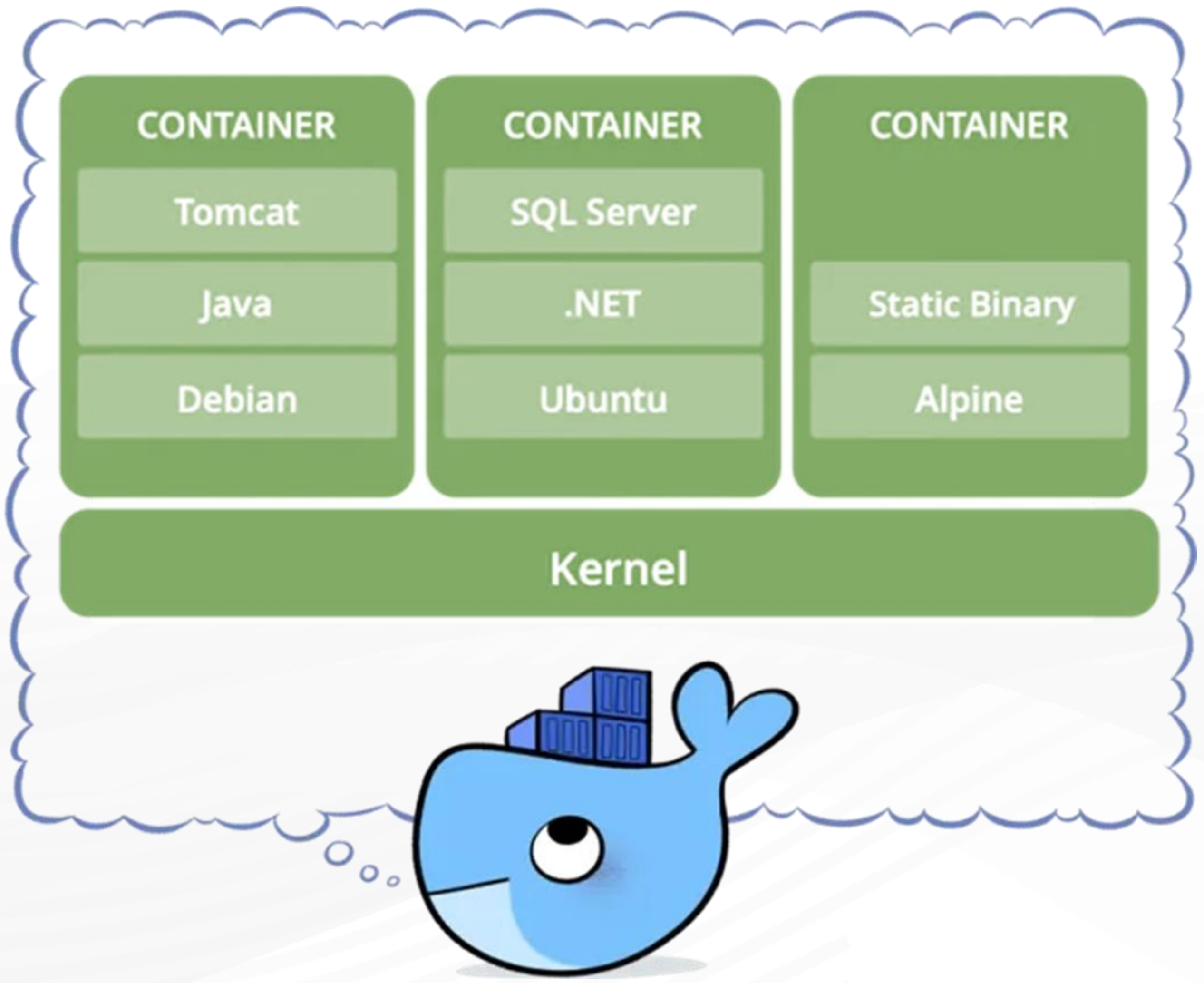
What did you Grasp?



1. Which technology helped LXC to stand out from the other container utilities?
   - **A) Root process can easily exit the chroot**
   - **B) Ability to isolate processes**
   - **C) Extraction at software level**
   - **D) C-groups**

1.23 Docker

In 2013, the first version of Docker was introduced. Developed by Google engineers collaborating with Docker over libcontainer and porting the core concepts and abstractions to libcontainer.



Source: https://i0.wp.com/www.docker.com/blog/wp-content/uploads/011f3ef6-d824-4d43-8b2c-36dab8eaaa72-1.jpg?fit=650%2C530&ssl=1

In a nutshell, we learnt:

1. Transporting "goods" analogy
2. Containerization platform, images and runtime
3. Comparison with virtual machines
4. The chroot system call
5. FreeBSD Jails
6. LinuX Containers (LXC)
7. Docker

End of Module

Next Module 2: Introduction to Containerization